Maximum Entropy Discrimination Markov Networks

Jun Zhu **Eric Xing** [a] **Bo Zhang**

February 2008
CMU-ML-08-104

[a]To whom correspondence should be addressed to.

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: Maximum entropy discrimination Markov networks, Bayesian max-margin Markov networks, Laplace max-margin Markov networks, Structured prediction

Abstract

Standard max-margin structured prediction methods concentrate directly on the input-output mapping, and the lack of an elegant probabilistic interpretation causes limitations. In this paper, we present a novel framework called *Maximum Entropy Discrimination Markov Networks* (MaxEntNet) to do Bayesian max-margin structured learning by using *expected* margin constraints to define a feasible distribution subspace and applying the maximum entropy principle to choose the best distribution from this subspace. We show that MaxEntNet subsumes the standard max-margin Markov networks (M^3N) as a spacial case where the predictive model is assumed to be linear and the parameter prior is a standard normal. Based on this understanding, we propose the Laplace max-margin Markov networks (LapM^3N) which use the Laplace prior instead of the standard normal. We show that the adoption of a Laplace prior of the parameter makes LapM^3N enjoy properties expected from a sparsified M^3N. Unlike the L_1-regularized maximum likelihood estimation which sets small weights to zeros to achieve sparsity, LapM^3N posteriorly weights the parameters and features with smaller weights are shrunk more. This posterior weighting effect makes LapM^3N more stable with respect to the magnitudes of the regularization coefficients and more generalizable. To learn a LapM^3N, we present an efficient iterative learning algorithm based on variational approximation and existing convex optimization methods employed in M^3N. The feasibility and promise of LapM^3N are demonstrated on both synthetic and real OCR data sets.

1 Introduction

In recent years, log-linear models based on composite features that explicitly exploit the structural dependencies among elements in high-dimensional inputs (e.g., DNA strings, text sequences, image lattices) and structured interpretational outputs (e.g., gene segmentation, natural language parsing, scene description) have gained substantial popularity in learning structured predictions from complex data. Major instances of such models include the conditional random fields (CRFs) [15], Markov networks (MNs) [25], and other specialized graphical models [1]. Adding to the flexibilities and expressive power of such models, different learning paradigms have been explored, such as maximum likelihood estimation [15], and max-margin learning [1, 25, 29].

While the probabilistic likelihood-based estimation defines a joint distribution of both input and output variables [22] or a conditional distribution of the output given the input [15], the standard max-margin structured prediction [25, 1, 29] takes the max-margin principle underlying support vector machines and concentrates directly on the input-output mapping. Although the max-margin principle could lead to a robust decision boundary, the lack of an elegant probabilistic interpretation causes limitations in standard max-margin structured learning. For example, it is not obvious how to consider missing data, such as in the learning of hidden hierarchical models [21, 34]. Another shortcoming, which is our focus in this paper, of the standard max-margin structured learning is that it is not easy to learn a "sparse" model.

For domains with complex feature space, it is often desirable to pursue a "sparse" representation of the model that leaves out irrelevant features. Learning such a sparse model is key to reduce the rick of over-fitting and achieve good generalization ability. In likelihood-based estimation, sparse model fitting has been extensively studied. A commonly used strategy is to add an L_1-penalty to the likelihood function, which can also be viewed as a MAP estimation under a Laplace prior. As noted by [14], the sparsity is due to a hard threshold introduced by the Laplace prior, and weights less than the threshold will be set to zeros. Recent work along this line includes [18, 30, 2].

In spite of recent advancements in likelihood-based estimation, little progress has been made so far on learning sparse MNs or log-linear models in general based on the max-margin principle, which is arguably a more desirable paradigm for training highly discriminative structured prediction models in a number of application contexts. While sparsity has been pursued in maximum margin learning of certain discriminative models such as SVM that are "unstructured" (i.e., with a univariate output), by using L_1-regularization [4] or by adding a cardinality constraint [6], generalization of these techniques to structured output space turns out to be extremely non-trivial. For example, although it appears possible to formulate sparse max-margin learning as a convex optimization problem as for SVM, both the primal and dual problems are hard to solve since there is no obvious way to exploit the conditional independence structures within a regularized Markov network to efficiently deal with the typically exponential number of constraints resulted from the max-margin condition. Another empirical insight as we will show in this paper is that the L_1-regularized estimation is not so robust. Discarding the features that are not completely irrelevant can

potentially hurt generalization ability.

In this paper, we propose a novel framework called *Maximum Entropy Discrimination Markov Networks* (MaxEntNet) to combine Bayesian learning and max-margin learning for structured prediction. MaxEntNet is a generalization of the maximum entropy discrimination [12] methods originally developed for single-label classification to the broader problem of structured learning. It facilitates posterior inference of a full distribution of feature coefficients (i.e., weights), rather than a point-estimate as in the standard max-margin Markov network (M^3N) [25], under a user-specified prior distribution of the coefficients and generalized maximum margin constraints. One can use the learned posterior distribution of coefficients to form a Bayesian max-margin Markov network that is equivalent to a weighted sum of differentially parameterized M^3Ns, or one can obtain a MAP M^3N. While the formalism of MaxEntNet is extremely general, we concentrate on a specialization that we denote the Laplace max-margin Markov networks ($LapM^3N$). We show that, by using a Laplace prior for the feature coefficients, the resulting $LapM^3N$ is effectively a "sparse" max-margin Markov network. But unlike the L_1-regularized maximum likelihood estimation, where sparsity is due to a hard threshold introduced by the Laplace prior [14], the effect of the Laplace prior in the $LapM^3N$ is a biased posterior weighting of the parameters. Smaller parameters are shrunk more and thus robust estimation is achieved when the data have irrelevant features. The Bayesian formalism also makes the $LapM^3N$ less sensitive to regularization constants. One of our interesting insights is that a trivial assumption on the prior distribution of the coefficients, i.e., a standard (zero-mean and identity covariance) normal, reduces the linear MaxEntNet to the standard M^3N, as shown in Theorem 3 in this paper. This understanding opens the way to use different priors in the Bayesian max-margin Markov networks. Our proposed $LapM^3N$ is a special case by using the Laplace prior.

To efficiently learn a Laplace M^3N, direct optimization can be very hard. Instead, we use the hierarchical representation of the Laplace prior [9] and develop a variational Bayesian method to efficiently learn the model. Based on existing convex optimization algorithms developed for M^3N [25, 3, 23], our learning algorithm is simple and easy to implement. It iteratively solves a QP problem, which is the same as that of the standard max-margin Markov networks, and updates a covariance matrix which is used in the QP problem. Note that in single label learning, sparse Bayesian learning and Relevance Vector Machine (RVM) [28] have been proposed to find a sparse solution for classification. But unlike SVM which directly optimizes margins, RVM defines a likelihood function from margins. Instead, we optimize a KL-divergence with a set of classification constraints that are explicitly defined with margins. This clarity makes it possible to develop a simple learning algorithm based on existing algorithms.

The rest of the paper is structured as follows. In the next section, we review the basic structured prediction formalism and set the stage for our model. Section 3 presents the maximum entropy discrimination Markov networks and some basic theoretical results. Section 4 presents the Laplace M^3N, and a novel iterative learning algorithm based on variational approximation and convex optimization. In Section 5, we briefly discuss the generalization bound of MaxEntNet. Then, we show empirical results on both synthetic and real OCR

data in Section 6. Section 7 discusses some related work and Section 8 concludes this paper.

2 Preliminaries

In a structured prediction problem, such as natural language parsing, image understanding, or DNA decoding, our objective is to learn a predictive function $h : \mathcal{X} \mapsto \mathcal{Y}$ from a structured input $\mathbf{x} \in \mathcal{X}$ (e.g., a sentence or an image) to a structured output $\mathbf{y} \in \mathcal{Y}$ (e.g., a sentence parsing or a scene annotation), where $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$ with $\mathcal{Y}_i = \{y_1, \ldots, y_{m_i}\}$ represents a combinatorial space of structured interpretations of multi-facet objects. For example, $\mathcal{Y}$ could correspond to the space of all possible instantiations of the part-of-speech (POS) tagging in the parse tree of a sentence, or the space of all possible ways of labeling entities over some segmentation of an image. The prediction $\mathbf{y} \equiv (y_1, \ldots, y_l)$ is *structured* because each individual label $y_i \in \mathcal{Y}_i$ within $\mathbf{y}$ must be determined in the context of other labels $y_{j \neq i}$, rather than independently as in a standard classification problem, in order to arrive at a globally satisfactory and consistent prediction.

Let $F : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ represent a discriminant function over the input-output pairs from which one can define the predictive function h. A common choice of F is a linear model, which is based on a set of feature functions $f_k : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ and their weights w_k, i.e., $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = g(\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}))$, where $\mathbf{f}$ is a K-dim column vector of the feature functions and $\mathbf{w}$ is the corresponding vector of the weights. Given F, the prediction function h is typically defined in terms of an optimization problem that maximizes F over the response variable $\mathbf{y}$ given input $\mathbf{x}$:

$$h_0(\mathbf{x}; \mathbf{w}) = \arg \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} F(\mathbf{x}, \mathbf{y}; \mathbf{w}), \tag{1}$$

where $\mathcal{Y}(\mathbf{x}) \in \mathcal{Y}$ is the feasible subset of structured labels for the sample $\mathbf{x}$. Here, we assume that for any sample $\mathbf{x}$, $\mathcal{Y}(\mathbf{x})$ is finite.

Depending on the specific choice of $F(\,\cdot\,; \mathbf{w})$ (e.g., linear, or log linear), and of the objective function $C(\mathbf{w})$ for estimating the parameter $\mathbf{w}$ (e.g., likelihood, or margin), incarnations of the general structured prediction formalism described above can be seen in models such as the CRFs [15], where $g(\cdot)$ is an exponential family conditional distribution function and $C(\mathbf{w})$ is the conditional likelihood of the true structured label; and the $\mathrm{M}^3\mathrm{N}$ [25], where $g(\cdot)$ is an identify function and $C(\mathbf{w})$ is the margin between the true label and any other label. Recent advances in structured prediction has introduced regularizations of $C(\mathbf{w})$ in the CRF context (i.e. likelihood-based estimation), so that a *sparse* $\mathbf{w}$ can be learned [2]. To the best of our knowledge, existing max-margin structured prediction methods utilize a single discriminant function $F(\,\cdot\,; \mathbf{w})$ defined by the "optimum" estimate of $\mathbf{w}$, similar to a practice in Frequentist statistics. Furthermore, the standard max-margin methods [25, 1, 29] concentrate directly on the input-output mapping and lacks an elegant probabilistic interpretation. This will cause limitations as discussed in the introduction, e.g., it is not obvious to incorporate missing data as in the learning of hidden hierarchical models [21, 34] and it is not easy to derive a "sparse" model. In this paper, we propose a Bayesian version of the predictive rule in Eq. (1) so that the prediction function h can be obtained from a posterior mean

over multiple (indeed infinitely many) $F(\,\cdot\,;\mathbf{w})$; and we also propose a new formalism and objective $C(\mathbf{w})$ that lead to a Bayesian M^{3}N, which subsumes the standard M^{3}N as a special case, and can achieve a posterior shrinkage effect on $\mathbf{w}$ that resembles L_1-regularization. To our knowledge, although sparse graphical model learning based on various likelihood-based principles has recently received substantial attention [18, 30], learning sparse networks based on the maximum margin principle has not yet been successfully explored. Our proposed method represents an initial foray in this important direction.

Before dwelling into exposition of the proposed approach, we end this section with a brief recapitulation of the basic M^{3}N that motivates this work, and provides a useful baseline that grounds the proposed approach. Under a max-margin framework, given a set of fully observed training data $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i \rangle\}_{i=1}^N$, we obtain a point estimate of the weight vector $\mathbf{w}$ by solving the following max-margin problem P0 [25]:

$$\begin{aligned} \text{P0 (M}^3\text{N)}: \quad & \min_{\mathbf{w},\xi} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t. } \forall i, \forall \mathbf{y} \neq \mathbf{y}^i: \quad & \mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i, \ \xi_i \geq 0 \ , \end{aligned}$$

where $\Delta\mathbf{f}_i(\mathbf{y}) = \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \mathbf{f}(\mathbf{x}^i, \mathbf{y})$ and $\mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y})$ is the "margin" between the true label $\mathbf{y}^i$ and a prediction $\mathbf{y}$, $\Delta\ell_i(\mathbf{y})$ is a loss function with respect to $\mathbf{y}^i$, and ξ_i represents a slack variable that absorbs errors in the training data. Various loss functions have been proposed in the literature [29]. In this paper, we adopt the *hamming loss* used in [25]: $\Delta\ell_i(\mathbf{y}) = \sum_{j=1}^{|\mathbf{x}^i|}\mathbb{I}(y_j \neq y_j^i)$, where $\mathbb{I}(\cdot)$ is an indicator function that equals to one if the argument is true and zero otherwise. The optimization problem P0 is intractable because the feasible space for $\mathbf{w}$, $\mathcal{F}_0 = \{\mathbf{w} : \mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i;\ \forall i, \forall \mathbf{y} \neq \mathbf{y}^i\}$, is defined by $O(N|\mathcal{Y}|)$ number of constraints, and $\mathcal{Y}$ itself is exponential to the size of the input $\mathbf{x}$. Exploring sparse dependencies among individual labels y_i in $\mathbf{y}$, as reflected in the specific design of the feature functions (e.g., based on pair-wise labeling potentials in a pair-wise Markov network), and the convex duality of the objective, efficient optimization algorithms based on cutting-plane [29] or message-passing [25] have been proposed to obtain an approximate optimum solution to P0. As described shortly, these algorithms can be directly employed as subroutines in solving our proposed model.

3 Maximum Entropy Discrimination Markov Networks

In this paper, we take a Bayesian approach and learn a distribution $p(\mathbf{w})$, rather than a point estimate of $\mathbf{w}$, in a max-margin manner. For prediction, we take the average over all the possible models, that is:

$$h_1(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \int p(\mathbf{w})F(\mathbf{x}, \mathbf{y}; \mathbf{w})\,\mathrm{d}\mathbf{w}\,. \tag{2}$$

Now, the open question is how we can devise an appropriate objective function over $p(\mathbf{w})$, in a similar spirit as the L_2-norm cost over $\mathbf{w}$ in P0, that leads to an optimum estimate of $p(\mathbf{w})$.

Below, we present a novel framework of *maximum entropy discrimination Markov networks* (MaxEntNet) that facilitates the estimation of a Bayesian M^{3}N defined by $p(\mathbf{w})$. As we show in the sequel, our Bayesian max-margin learning formalism offers several advantages like the PAC-Bayes generalization guarantee and estimation robustness.

3.1 The Basic MED

The basic maximum entropy discrimination (MED) [12] framework is studied for single-label learning, where the output consists of only one class label. For example, for the single-label binary classification, the training data are $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^N$ where x^i is a feature vector and $y^i \in \{+1, -1\}$. In MED, the prediction rule is:

$$\hat{y} = \text{sign} \int p(\mathbf{w}) F(x, y; \mathbf{w}) \, \mathrm{d}\mathbf{w} \,,$$

where the discriminant function can be a linear function $\mathbf{w}^\top x + b$ or the general log-likelihood ratio of two generative models: $\log \frac{p(x|\mathbf{w}_+)}{p(x|\mathbf{w}_-)} + b$. Here, b is a scalar bias term. To find the best distribution $p(\mathbf{w})$, MED solves the following optimization problem:

$$\begin{aligned} \min_{p(\Theta), \xi} \quad & KL(p(\Theta)||p_0(\Theta)) \\ \text{s.t.} \quad & \int p(\Theta)[y_i F(x, y; \mathbf{w}) - \xi_i] \, \mathrm{d}\Theta \; \geq 0, \forall i, \end{aligned}$$

where Θ can be the model parameter $\mathbf{w}$ when ξ are kept fixed or the pair of model parameter and slack variable $(\mathbf{w}, \xi)$ when we want to optimize over ξ.

3.2 MaxEntNet and the Bayesian M^{3}N

Given a training set $\mathcal{D}$ of structured input-output pairs, analogous to the feasible space $\mathcal{F}_0$ for weight vector $\mathbf{w}$ in the standard M^{3}N (i.e., problem P0), the feasible subspace $\mathcal{F}_1$ of weight distribution $p(\mathbf{w})$ is defined by a set of *expected* margin constraints:

$$\mathcal{F}_1 = \big\{ p(\mathbf{w}) : \int p(\mathbf{w}) [\Delta F_i(\mathbf{y}; \mathbf{w}) - \Delta \ell_i(\mathbf{y})] \, \mathrm{d}\mathbf{w} \geq -\xi_i, \; \forall i, \forall \mathbf{y} \neq \mathbf{y}^i \big\},$$

where $\Delta F_i(\mathbf{y}; \mathbf{w}) = F(\mathbf{x}^i, \mathbf{y}^i; \mathbf{w}) - F(\mathbf{x}^i, \mathbf{y}; \mathbf{w})$.

To choose the best distribution $p(\mathbf{w})$ from $\mathcal{F}_1$, the *maximum entropy principle* suggests that one can consider the distribution that minimizes its relative entropy with respect to some chosen prior p_0, as measured by the Kullback-Leibler divergence, $KL(p||p_0) = \langle \log(p/p_0) \rangle_p$, where $\langle \cdot \rangle_p$ denotes the expectations with respect to p. If p_0 is uniform, then minimizing the KL-divergence is equivalent to maximizing the entropy $H(p) = -\langle \log p \rangle_p$. To accommodate the discriminative prediction problem we concern, instead of minimizing the usual KL, we optimize the generalized entropy [7, 17], or a regularized KL-divergence, $KL(p(\mathbf{w})||p_0(\mathbf{w})) + U(\xi)$, where $U(\xi)$ is a closed proper convex function over the slack variables and it is also known as an additional "potential" term in the maximum entropy principle. This leads to the following Maximum Entropy Discrimination Markov Networks:

Definition 1 (Maximum Entropy Discrimination Markov Networks) *Given training data* $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i \rangle\}_{i=1}^N$*, a discriminant function* $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$*, a loss function* $\Delta\ell_{\mathbf{x}}(\mathbf{y})$*, and an ensuing feasible subspace* $\mathcal{F}_1$ *(defined above) for parameter distribution* $p(\mathbf{w})$*, the MaxEntNet model that leads to a prediction function of the form of Eq. (2) is defined by the following generalized relative entropy minimization with respect to a parameter prior* $p_0(\mathbf{w})$*:*

$$\begin{aligned} \text{P1}: \quad & \min_{p(\mathbf{w}),\xi} \; KL(p(\mathbf{w})||p_0(\mathbf{w})) + U(\xi) \\ & \text{s.t.} \;\; p(\mathbf{w}) \in \mathcal{F}_1, \;\; \xi_i \geq 0, \forall i. \end{aligned}$$

The P1 defined above is a variational optimization problem over $p(\mathbf{w})$ in a subspace of valid parameter distributions. Since both the KL and the function U in P1 are convex, and the constraints in $\mathcal{F}_1$ are linear, P1 is a convex program, which can be solved via applying the calculus of variations to the Lagrangian to obtain a variational extremum, followed by a dual transformation of P1. We state the main results as a theorem.

Theorem 2 (Solution to MaxEntNet) *The variational optimization problem P1 underlying the MaxEntNet gives rise to the following optimum distribution of Markov network parameters* $\mathbf{w}$*:*

$$p(\mathbf{w}) = \frac{1}{Z(\alpha)} p_0(\mathbf{w}) \exp\big\{ \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) [\Delta F_i(\mathbf{y}; \mathbf{w}) - \Delta\ell_i(\mathbf{y})] \big\}, \tag{3}$$

where $Z(\alpha)$ *is a normalization factor and the Lagrangian multipliers* $\alpha_i(\mathbf{y})$ *(corresponding to constraints in* $\mathcal{F}_1$*) can be obtained by solving the dual problem of P1:*

$$\begin{aligned} \text{D1}: \quad & \max_{\alpha} \;\; -\log Z(\alpha) - U^\star(\alpha) \\ & \text{s.t.} \;\; \alpha_i(\mathbf{y}) \geq 0, \;\; \forall i, \;\; \forall \mathbf{y}, \end{aligned}$$

where $U^\star(\cdot)$ *represents the conjugate of the slack function* $U(\cdot)$*, i.e.,* $U^\star(\alpha) = \sup_\xi \big(\sum_{i,\mathrm{y}} \alpha_i(\mathrm{y}) \xi_i - U(\xi) \big)$*.*

Proof: Since both the KL-divergence and U are convex and the constraints are linear, the problem P1 is a convex program. To compute the convex dual, we introduce a non-negative dual variable $\alpha_i(\mathbf{y})$ for each constraint in $\mathcal{F}_1$ and a non-negative variable c for the normalization constraint $\int p(\mathbf{w})\, \mathrm{d}\mathbf{w} = 1$. Then, we form the Lagrangian as,

$$\begin{aligned} L(p(\mathbf{w}), \xi, \alpha, c) = & \; KL(p(\mathbf{w})||p_0(\mathbf{w})) + U(\xi) \\ & - \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \big(\int p(\mathbf{w}) [\Delta F_i(\mathbf{y}; \mathbf{w}) - \Delta\ell_i(\mathbf{y})]\, \mathrm{d}\mathbf{w} + \xi_i \big) + c \big(\int p(\mathbf{w})\, \mathrm{d}\mathbf{w} - 1 \big). \end{aligned}$$

The Lagrangian dual function is defined as $L^\star(\alpha, c) \triangleq \inf_{p(\mathbf{w});\xi} L(p(\mathbf{w}), \xi, \alpha, c)$. Take the derivative of L w.r.t $p(\mathbf{w})$, then we get,

$$\frac{\partial L}{\partial p(\mathbf{w})} = 1 + c + \log \frac{p(\mathbf{w})}{p_0(\mathbf{w})} - \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta \ell_i(\mathbf{y})].$$

Set the derivative to zero and we get the distribution $p(\mathbf{w})$,

$$p(\mathbf{w}) = \frac{1}{Z(\alpha)} p_0(\mathbf{w}) \exp\big\{ \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]\big\},$$

where $Z(\alpha) \triangleq \int p_0(\mathbf{w}) \exp\big\{ \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]\big\}\, \mathrm{d}\mathbf{w}$ is a normalization constant and $c = -1 - \log Z(\alpha)$.

Substitute $p(\mathbf{w})$ into $L^\star$, and we get,

$$\begin{aligned} L^\star(\alpha, c) &= \inf_{p(\mathbf{w});\xi} \big(-\log Z(\alpha) + U(\xi) - \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\xi_i \big) \\ &= -\log Z(\alpha) + \inf_{\xi} \big(U(\xi) - \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\xi_i \big) \\ &= -\log Z(\alpha) - \sup_{\xi} \big(\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\xi_i - U(\xi) \big). \end{aligned}$$

Let $\alpha' = (\alpha_1, \ldots, \alpha_N)^\top$ and $\alpha_i = \sum_{\mathbf{y}} \alpha_i(\mathbf{y})$. Then, the second term on the right hand side of the above last equation is $\sup_\xi((\alpha')^\top \xi - U(\xi))$. This is the definition of the conjugate of $U(\xi)$. Let $U^\star$ be the conjugate of U, and we get,

$$L^\star(\alpha, c) = -\log Z(\alpha) - U^\star(\alpha').$$

Without causing ambiguity, we use α instead of α'. Now, we get the dual problem D1. □

For a closed proper convex function $\phi(\mu)$, its conjugate is defined as $\phi^\star(\nu) = \sup_\mu[\nu^\top \mu - \phi(\mu)]$. In problem D1, by convex duality [5], the log normalizer $\log Z(\alpha)$ can be shown to be the conjugate of the KL-divergence. If the slack function is $U(\xi) = C\|\xi\| = C\sum_i \xi_i$, it is easy to show that $U^\star(\alpha) = \mathbb{I}_\infty(\sum_{\mathbf{y}} \alpha_i(\mathbf{y}) \le C,\ \forall i)$, where $\mathbb{I}_\infty(\cdot)$ is a function that equals to zero when its argument holds true and infinity otherwise. Here, the inequality corresponds to the trivial solution $\xi = 0$, that is, the training data are perfectly separative. Ignoring this inequality does not affect the solution since the special case $\xi = 0$ is still included. Thus, the Lagrangian multipliers $\alpha_i(\mathbf{y})$ in the dual problem D1 comply with the set of constraints that $\sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C,\ \forall i$. Another example is $U(\xi) = KL(p(\xi)||p_0(\xi))$ by introducing uncertainty on the slack variables [12]. In this case, expectations with respect to $p(\xi)$ are taken on both sides of all the constraints in $\mathcal{F}_1$. Take the duality, and the dual function of U is another log normalizer. More details can be found in [12]. Some other U functions and their dual functions are studied in [17, 7].

The MaxEntNet model gives an optimum parameter distribution, which is used to make prediction via the rule (2). An alternative way to understand our proposed model is suggested

by the striking isomorphisms of the opt-problem P1, the feasible space $\mathcal{F}_1$, and the predictive function h_1 underlying a MaxEntNet, to their counterparts P0, $\mathcal{F}_0$, and h_0, respectively, underlying an M^{3}N. Indeed, by making a special choice of the parameter prior in Eq. (3), based on the above discussion of conjugate functions in D1, we arrive at a reduction of D1 to an M^{3}N optimization problem. Thus, we also call the MaxEntNet a Bayesian M^{3}N (BM^{3}N). The following theorem makes this explicit.

Theorem 3 (Reduction of MaxEntNet to M^{3}N) *Assuming $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$, $U(\xi) = \sum_i \xi_i$, and $p_0(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, I)$, where I denotes an identity matrix, then the Lagrangian multipliers $\alpha_i(\mathbf{y})$ are obtained by solving the following dual problem:*

$$\begin{aligned} \max_{\alpha} \ & \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\ell_i(\mathbf{y}) - \frac{1}{2} \| \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\mathbf{f}_i(\mathbf{y}) \|^2 \\ \text{s.t.} \ & \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C; \ \alpha_i(\mathbf{y}) \geq 0, \ \forall i, \ \forall \mathbf{y}, \end{aligned}$$

which, when applied to h_1, lead to a predictive function that is identical to $h_0(\mathbf{x}; \mathbf{w})$ given by Eq. (1).

Proof: Replacing $p_0(\mathbf{w})$ and $\Delta F_i(\mathbf{y}; \mathbf{w})$ in Eq. (3) with $\mathcal{N}(\mathbf{w}|0, I)$ and $\mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y})$ respectively, we can obtain the following closed-form expression of the $Z(\alpha)$ in $p(\mathbf{w})$:

$$\begin{aligned} Z(\alpha) &\triangleq \int \mathcal{N}(\mathbf{w}|0, I) \exp\Big\{ \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) [\mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) - \Delta\ell_i(\mathbf{y})] \Big\} \, \mathrm{d}\mathbf{w} \\ &= \int (2\pi)^{-\frac{K}{2}} \exp\Big\{ -\frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) [\mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) - \Delta\ell_i(\mathbf{y})] \Big\} \, \mathrm{d}\mathbf{w} \\ &= \exp\Big(-\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\ell_i(\mathbf{y}) + \frac{1}{2} \| \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\mathbf{f}_i(\mathbf{y}) \|^2 \Big). \end{aligned}$$

As we have stated, the constraints $\sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C$ are due to the conjugate of $U(\xi) = \sum_i \xi_i$.

For prediction, again replacing $p_0(\mathbf{w})$ and $\Delta F_i(\mathbf{y}; \mathbf{w})$ in Eq. (3) with $\mathcal{N}(\mathbf{w}|0, I)$ and $\mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y})$ respectively, we can get $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mu_{\mathbf{w}}, I)$, where $\mu_{\mathbf{w}} = \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\mathbf{f}_i(\mathbf{y})$. Substituting $p(\mathbf{w})$ into the predictive function h_1, we can get $h_1(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mu_{\mathbf{w}}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = (\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta\mathbf{f}_i(\mathbf{y}))^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$, which is identical to the prediction rule of the standard M^{3}N [25]. □

Theorem 3 shows that in the supervised learning setting, the M^{3}Ns are subsumed by the MaxEntNet model, and can be viewed as a special case of a Bayesian M^{3}N when the slack function is linear and the parameter prior is a standard normal. As we shall see later, this connection renders many existing techniques for solving the M^{3}N directly applicable for solving the MaxEntNet or BM^{3}N.

Recent trend in pursuing "sparse" graphical models has led to the emergence of regularized version of CRFs [2] and Markov networks [18, 30]. Interestingly, while such extensions have been successfully implemented by several authors in maximum likelihood learning of various sparse graphical models, they have not yet been explored in the context of maximum

margin learning. Such a gap is not merely due to a negligence. Indeed, learning a sparse M^{3}N can be significantly harder as we discuss below.

As Theorem 3 reveals, an M^{3}N corresponds to a BM^{3}N with a standard normal prior for the weight vector $\mathbf{w}$. To encourage a sparse model, when using zero-mean normal prior, the weights of irrelevant features should peak around zero with very small variances. However, the isotropy of the variances in all dimensions in the standard normal prior makes M^{3}N infeasible to adjust the variances in different dimensions to fit sparse data. One way to learn a sparse model is to adopt the strategy of L_1-SVM [4, 33] to use L_1-norm instead of L_2-norm (see appendix B for a detailed description of this formulation and the duality derivation). However, in both the primal and dual of an L_1-regularized M^{3}N, there is no obvious way to exploit the sparse dependencies among variables of the Markov network in order to efficiently deal with typically exponential number of constraints, which makes direct optimization or LP-formulation expensive. In this paper, we adopt the MaxEntNet framework that directly leads to a Bayesian M^{3}N, and employ a Laplace prior for $\mathbf{w}$ to learn a Laplace M^{3}N. When fitted to training data, the parameter posterior $p(\mathbf{w})$ under a Laplace M^{3}N has a shrinkage effect on small weights, which is similar to the L_1-regularizer in an M^{3}N. Although exact learning of a Laplace M^{3}N is still very hard, we show that it can be efficiently approximated by a variational inference procedure based on existing methods.

4 Laplace M^{3}N

The Laplace prior of $\mathbf{w}$ is $p_0(\mathbf{w}) = \prod_{k=1}^K \frac{\sqrt{\lambda}}{2} e^{-\sqrt{\lambda}|w_k|} = \left(\frac{\sqrt{\lambda}}{2}\right)^K e^{-\sqrt{\lambda}\|\mathbf{w}\|}$. The Laplace density is heavy tailed and peaked at zero. Thus, it encodes the prior belief that the distribution of $\mathbf{w}$ is strongly peaked around zero. Another nice property is that the Laplace density is log-convex, which can be exploited to get convex estimation problems like LASSO [27].

4.1 Variational Learning with Laplace Prior

Although in principle we have a closed-form solution of $p(\mathbf{w})$ in Theorem 2, the parameters $\alpha_i(\mathbf{y})$ are hard to estimate when using the Laplace prior. As we shall see in Section 4.2, exact integration will lead to a dual function that is difficult to maximize. Thus, we present a variational approximate learning approach.

Our approach is based on the hierarchical interpretation [9] of the Laplace prior, that is, each w_k has a zero-mean Gaussian distribution $p(w_k|\tau_k) = \mathcal{N}(w_k|0, \tau_k)$ and the variance τ_k has an exponential hyper-prior density,

$$p(\tau_k|\lambda) = \frac{\lambda}{2} \exp\big\{-\frac{\lambda}{2}\tau_k\big\}, \quad \text{for } \tau_k \geq 0.$$

Then, we have $p_0(\mathbf{w}) = \prod_{k=1}^K p_0(w_k) = \prod_{k=1}^K \int p(w_k|\tau_k)p(\tau_k|\lambda)\,\mathrm{d}\tau_k = \int p(\mathbf{w}|\tau)p(\tau|\lambda)\,\mathrm{d}\tau$, where $p(\mathbf{w}|\tau) = \prod_{k=1}^K p(w_k|\tau_k)$ and $p(\tau|\lambda) = \prod_{k=1}^K p(\tau_k|\lambda)$ are joint distributions and $\mathrm{d}\tau \triangleq \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_K$. Using the hierarchical representation of the Laplace prior and applying the Jensen's inequality, we get an upper bound of the KL-divergence,

Algorithm 1 Variational Bayesian Learning

Input: data $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i \rangle\}_{i=1}^N$, constants C and λ, iteration number T
Output: posterior mean $\langle \mathbf{w} \rangle_p^T$
Initialize $\langle \mathbf{w} \rangle_p^1 \leftarrow 0$, $\Sigma_{\mathbf{w}}^1 \leftarrow I$
for $t = 1$ **to** $T - 1$ **do**
  Step 1: solve (5) or (6) for $\langle \mathbf{w} \rangle_p^{t+1} = \Sigma_{\mathbf{w}}^t \eta$; update $\langle \mathbf{w}\mathbf{w}^\top \rangle_p^{t+1} \leftarrow \Sigma_{\mathbf{w}}^t + \langle \mathbf{w} \rangle_p^{t+1} (\langle \mathbf{w} \rangle_p^{t+1})^\top$.
  Step 2: use (7) to update $\Sigma_{\mathbf{w}}^{t+1} \leftarrow \text{diag}(\sqrt{\frac{\langle w_k^2 \rangle_p^{t+1}}{\lambda}})$.
end for

$$
\begin{aligned}
KL(p||p_0) &= -H(p) - \langle \log \int p(\mathbf{w}|\tau) p(\tau|\lambda) \, \mathrm{d}\tau \, \rangle_p \\
&\leq -H(p) - \langle \int q(\tau) \log \frac{p(\mathbf{w}|\tau) p(\tau|\lambda)}{q(\tau)} \, \mathrm{d}\tau \, \rangle_p \\
&\triangleq \mathcal{L}(p(\mathbf{w}), q(\tau)),
\end{aligned}
$$

where $q(\tau)$ is a variational distribution which is used to approximate $p(\tau|\lambda)$.

Substituting this upper bound for the KL in P1, we now solve the following problem,

$$
\min_{p(\mathbf{w}) \in \mathcal{F}_1; q(\tau); \xi} \; \mathcal{L}(p(\mathbf{w}), q(\tau)) + U(\xi). \tag{4}
$$

This problem can be solved with an iterative minimization algorithm alternating between $p(\mathbf{w})$ and $q(\tau)$, as outlined in Algorithm 1, and detailed below.

Step 1: Keep $q(\tau)$ fixed, we optimize (4) with respect to $p(\mathbf{w})$. Taking the same procedure as in solving P1, we get the posterior distribution $p(\mathbf{w})$ as follows,

$$
\begin{aligned}
p(\mathbf{w}) &\propto \exp\{ \int q(\tau) \log p(\mathbf{w}|\tau) \, \mathrm{d}\tau \; - b\} \cdot \exp\{\mathbf{w}^\top \eta - L\} \\
&\propto \exp\{-\frac{1}{2} \mathbf{w}^\top \langle A^{-1} \rangle_q \mathbf{w} - b + \mathbf{w}^\top \eta - L\} \\
&= \mathcal{N}(\mathbf{w}|\mu_{\mathbf{w}}, \Sigma_{\mathbf{w}}),
\end{aligned}
$$

where $\eta = \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta \mathbf{f}_i(\mathbf{y})$, $L = \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta \ell_i(\mathbf{y})$, $A = \text{diag}(\tau_k)$, and $b = KL(q(\tau)||p(\tau|\lambda))$ is a constant. The posterior mean and variance are $\langle \mathbf{w} \rangle_p = \mu_{\mathbf{w}} = \Sigma_{\mathbf{w}} \eta$ and $\Sigma_{\mathbf{w}} = (\langle A^{-1} \rangle_q)^{-1} = \langle \mathbf{w}\mathbf{w}^\top \rangle_p - \langle \mathbf{w} \rangle_p \langle \mathbf{w} \rangle_p^\top$, respectively. The dual parameters α are estimated by solving the following dual problem:

$$
\begin{aligned}
\max_{\alpha} \;\; & \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y}) \Delta \ell_i(\mathbf{y}) - \frac{1}{2} \eta^\top \Sigma_{\mathbf{w}} \eta \\
\text{s.t.} \;\; & \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C; \;\; \alpha_i(\mathbf{y}) \geq 0, \;\; \forall i, \;\; \forall \mathbf{y}.
\end{aligned} \tag{5}
$$

This dual problem can be directly solved using existing algorithms developed for M^{3}N, such as [25, 3]. Alternatively, we can solve the following primal problem:

$$\begin{aligned} \min_{\mathbf{w},\xi} \ & \frac{1}{2}\mathbf{w}^\top \Sigma_{\mathbf{w}}^{-1}\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \ & \mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i;\ \xi_i \geq 0, \quad \forall i,\ \forall \mathbf{y} \neq \mathbf{y}^i. \end{aligned} \tag{6}$$

It is easy to show that the solution of problem (6) leads to the posterior mean of $\mathbf{w}$ under $p(\mathbf{w})$. Since $p(\mathbf{w})$ is a normal distribution, the posterior mean is the only parameter that is needed to do prediction by h_1. The primal problem can be solved with subgradient [23] or extragradient [26] methods.

Step 2: Keep $p(\mathbf{w})$ fixed, we optimize (4) with respect to $q(\tau)$. Take the derivative of $\mathcal{L}$ with respect to $q(\tau)$ and set it to zero, then we get.

$$q(\tau) \propto p(\tau|\lambda)\exp\left\{\langle \log p(\mathbf{w}|\tau)\rangle_p\right\}.$$

By exploring the factorization forms of $p(\mathbf{w}|\tau)$ and $p(\tau|\lambda)$, we can get an induced factorization $q(\tau) = \prod_{k=1}^{K} q(\tau_k)$ and each $q(\tau_k)$ is computed as follows:

$$\begin{aligned} \forall k: \quad q(\tau_k) &\propto p(\tau_k|\lambda)\exp\left\{\langle \log p(w_k|\tau_k)\rangle_p\right\} \\ &\propto \mathcal{N}(\sqrt{\langle w_k^2\rangle_p}|0,\tau_k)\exp(-\frac{1}{2}\lambda\tau_k). \end{aligned}$$

The same distribution has been derived in [14], and similarly we can get the normalization factor: $\int \mathcal{N}(\sqrt{\langle w_k^2\rangle_p}|0,\tau_k)\cdot\frac{\lambda}{2}\exp(-\frac{1}{2}\lambda\tau_k)\,\mathrm{d}\tau_k = \frac{\sqrt{\lambda}}{2}\exp(-\sqrt{\lambda\langle w_k^2\rangle_p})$. Also, as in [14], we can calculate the expectations $\langle\tau_k^{-1}\rangle_q$ which are required in calculating $\langle A^{-1}\rangle_q$ as follows,

$$\langle\frac{1}{\tau_k}\rangle_q = \int \frac{1}{\tau_k}q(\tau_k)\,\mathrm{d}\tau_k = \sqrt{\frac{\lambda}{\langle w_k^2\rangle_p}}. \tag{7}$$

We iterate between the above two steps until convergence. Then, we use the posterior distribution $p(\mathbf{w})$, which is a normal distribution, to make prediction. For irrelevant features, the variances should converge to zeros and thus lead to a sparse estimation. The intuition behind this iterative minimization algorithm is as follows. First, we use a Gaussian distribution to approximate the Laplace distribution and thus get a QP problem that is analogous to that of the standard M^{3}N; then, the second step updates the covariance matrix in the QP problem with an exponential hyper-prior on the variance.

4.2 Insights

To see how the Laplace prior affects the posterior distribution, we examine the posterior mean via an exact integration as follows.

Substitute the hierarchical representation of the Laplace prior into $p(\mathbf{w})$ in Theorem 2, and we get the normalization factor $Z(\alpha)$ as follows,

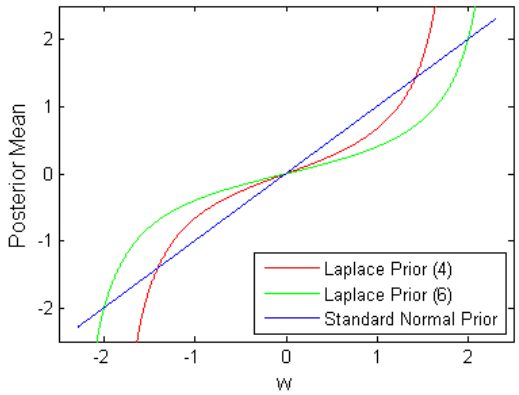

Figure 1: Posterior mean with different priors against the estimation of M^{3}N (i.e. with the standard normal prior).

$$
\begin{aligned}
Z(\alpha) &= \int\int p(\mathbf{w}|\tau)p(\tau|\lambda)\,\mathrm{d}\tau \cdot \exp\{\mathbf{w}^\top\eta - L\}\,\mathrm{d}\mathbf{w} \\
&= \int p(\tau|\lambda)\int p(\mathbf{w}|\tau)\cdot\exp\{\mathbf{w}^\top\eta - L\}\,\mathrm{d}\mathbf{w}\;\mathrm{d}\tau \\
&= \int p(\tau|\lambda)\int \mathcal{N}(\mathbf{w}|0,A)\exp\{\mathbf{w}^\top\eta - L\}\,\mathrm{d}\mathbf{w}\;\mathrm{d}\tau \\
&= \int p(\tau|\lambda)\exp\{\frac{1}{2}\eta^\top A\eta - L\}\mathrm{d}\tau \\
&= \exp\{-L\}\prod_{k=1}^{K}\int \frac{\lambda}{2}\exp(-\frac{\lambda}{2}\tau_k)\exp(\frac{1}{2}\eta_k^2\tau_k)\mathrm{d}\tau_k \\
&= \exp\{-L\}\prod_{k=1}^{K}\frac{\lambda}{\lambda-\eta_k^2},
\end{aligned} \tag{8}
$$

where $\eta_k = \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})(f_k(\mathbf{x}^i,\mathbf{y}^i) - f_k(\mathbf{x}^i,\mathbf{y}))$ and the last equality is due to the moment generating function of an exponential distribution. An additional constraint is $\eta_k^2 < \lambda,\ \forall k$. Otherwise, the integration is infinity. Using the integration result, we can get:

$$
\frac{\partial \log Z}{\partial \alpha_i(\mathbf{y})} = \mu^\top\Delta\mathbf{f}_i(\mathbf{y}) - \Delta\ell_i(\mathbf{y}), \tag{9}
$$

where μ is a column vector and $\mu_k = \frac{2\eta_k}{\lambda-\eta_k^2},\ \forall 1 \le k \le K$. An alternative way to compute the derivatives is using the definition of Z: $Z = \int p_0(\mathbf{w})\cdot\exp\{\mathbf{w}^\top\eta - L\}\,\mathrm{d}\mathbf{w}$. We can get:

$$
\frac{\partial \log Z}{\partial \alpha_i(\mathbf{y})} = \langle\mathbf{w}\rangle_p^\top\Delta\mathbf{f}_i(\mathbf{y}) - \Delta\ell_i(\mathbf{y}). \tag{10}
$$

Comparing Eqs. (9) and (10), we get $\langle\mathbf{w}\rangle_p = \mu$, that is, $\langle w_k\rangle_p = \frac{2\eta_k}{\lambda-\eta_k^2},\ \forall 1 \le k \le K$. Similar calculation can lead to the result that in the standard M^{3}N (i.e. with the standard normal prior) the posterior mean is $\langle\mathbf{w}\rangle_p = \eta$. As shown in [25], η is the optimal point estimate of M^{3}N. Figure 1 shows the posterior means (for any dimension) when the priors

are standard normal, Laplace with $\lambda = 4$, and Laplace with $\lambda = 6$ against the optimal point estimate of the standard M^{3}N. We can see that with a Laplace prior, the parameters are shrunk around zero. The larger the λ value is, the greater the shrinkage effect. For a fixed λ, the shape of the posterior mean is smoothly nonlinear but no component is explicitly discarded, that is, no weight is set to zero. This is different from the shape of a L_1-regularized maximum likelihood estimation [14] where an interval exists around the origin and parameters falling into this interval are set to zeros.

Note that if we use the exact integration as in Eq. (8), the dual problem D1 will maximize $L - \sum_{k=1}^{K} \log \frac{\lambda}{\lambda - \eta_k^2}$. Since η_k^2 appears within a logarithm, the optimization problem would be very hard to solve. Thus, we turn to a variational approximation method.

5 Generalization Bound

The PAC-Bayes bound [16] provides a theoretical motivation to learn an averaging model as in P1 which minimizes the KL-divergence and simultaneously satisfies the discriminative classification constraints. To apply it in our structured learning setting, we assume that all the discriminant functions are bounded, that is, there exists a positive constant c: $F(\,\cdot\,;\mathbf{w}) \in \mathcal{H} : \mathcal{X} \times \mathcal{Y} \to [-c, c],\ \forall \mathbf{w}$. Recall that our averaging model is defined as $h(\mathbf{x}, \mathbf{y}) = \langle F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \rangle_{p(\mathbf{w})}$. We define the margin of an example $(\mathbf{x}, \mathbf{y})$ for such a function h as $M(h, \mathbf{x}, \mathbf{y}) = h(\mathbf{x}, \mathbf{y}) - \max_{\mathbf{y}' \neq \mathbf{y}} h(\mathbf{x}, \mathbf{y}')$. Clearly, the model h makes a wrong prediction on $(\mathbf{x}, \mathbf{y})$ only if $M(h, \mathbf{x}, \mathbf{y}) \leq 0$. Let Q be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\mathcal{D}$ be a sample of N examples randomly drawn from Q. With these definitions, we have the following PAC-Bayes theorem.

Theorem 4 (PAC-Bayes Bound of MaxEntNet) *Let p_0 be any continuous probability distribution over $\mathcal{H}$ and let $\delta \in (0, 1)$. If $\forall \mathbf{w},\ F(\,\cdot\,;\mathbf{w}) \in \mathcal{H} : \mathcal{X} \times \mathcal{Y} \to [-c, c]$, then with probability at least $1 - \delta$ over random samples $\mathcal{D}$ of Q, for very distribution p over $\mathcal{H}$ and for all margin thresholds $\gamma > 0$:*

$$\Pr_Q(M(h, \mathbf{x}, \mathbf{y}) \leq 0) \leq \Pr_{\mathcal{D}}(M(h, \mathbf{x}, \mathbf{y}) \leq \gamma) + O\Big(\sqrt{\frac{\gamma^{-2} KL(p||p_0) \ln(N|\mathcal{Y}|) + \ln N + \ln \delta^{-1}}{N}}\Big).$$

Here, $\Pr_Q(.)$ stands for $\langle . \rangle_Q$ and $\Pr_{\mathcal{D}}(.)$ stands for the empirical average on $\mathcal{D}$. The proof follows the same structure as the proof of the original PAC-Bayes bound, with consideration of the margins. See appendix A for the details.

6 Experiments

In this section, we present some empirical results of the proposed Laplace max-margin Markov networks on both synthetic and real data sets. We compare LapM^{3}N with M^{3}N, CRFs, L_1-regularized CRFs (L_1-CRFs), and L_2-regularized CRFs (L_2-CRFs). We use the quasi-Newton method and its variant [2] to solve the optimization problem of CRFs, L_1-CRFs, and L_2-CRFs. For M^{3}N and LapM^{3}N, we can use the exponentiated gradient method

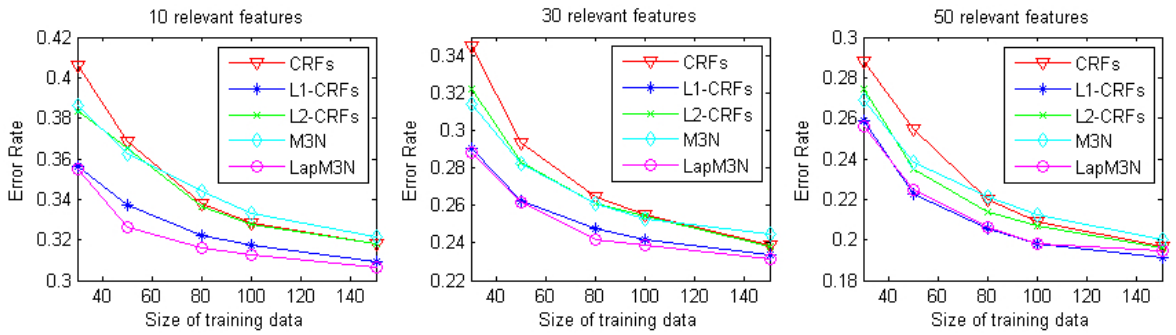

Figure 2: Evaluation results on data sets with i.i.d features.

[3] or structured minimal optimization [25] to solve the dual QP problem or solve a primary problem by using sub-gradient methods [23].

6.1 Synthetic Data Sets

We present some empirical results on synthetic data sets with ideally i.i.d features and data sets with more real correlated features.

6.1.1 I.I.D Features

The first experiment is conducted on synthetic sequence data with 100 i.i.d features. We generate three types of data sets with 10, 30, and 50 relevant features respectively. For each setting, we randomly generate 10 linear-chain CRFs with 8 binary labeling states. The feature functions include: a real valued state-feature function over a one dimensional input feature and a class label; and 4 (2×2) binary transition-feature functions capturing pairwise label dependencies. For each model we generate a data set of 1000 samples. For each sample, we first *independently* draw the 100 features from a standard normal distribution, and then apply a Gibbs sampler to assign a label sequence with 5000 iterations.

For each data set, we randomly draw a part as training data and use the rest for testing. The numbers of training data are 30, 50, 80, 100, and 150. The QP problem is solved with the exponentiated gradient method [3]. In all the following experiments, the regularization constant of L_1-CRFs and L_2-CRFs is chosen from $\{0.01, 0.1, 1, 4, 9, 16\}$ by a 5-fold cross-validation during the training. For the LapM^{3}N, we use the same method to choose λ from 20 roughly evenly spaced values between 1 and 268. For each setting, the average over 10 data sets is the final performance.

The results are shown in Figure 2. All the results of the LapM^{3}N are achieved with 3 iterations of the variational Bayesian learning. Under different settings LapM^{3}N consistently outperforms M^{3}N and performs comparably with the sparse L_1-CRFs. But note that the synthetic data come from simulated CRFs. Both the L_1-CRFs and L_2-CRFs outperform the un-regularized CRFs. One interesting result is that the M^{3}N and L_2-CRFs perform comparably. This is reasonable because as derived by [17] and noted by [11] the L_2-regularized maximum likelihood estimation of CRFs has a similar convex dual as that of the M^{3}N.

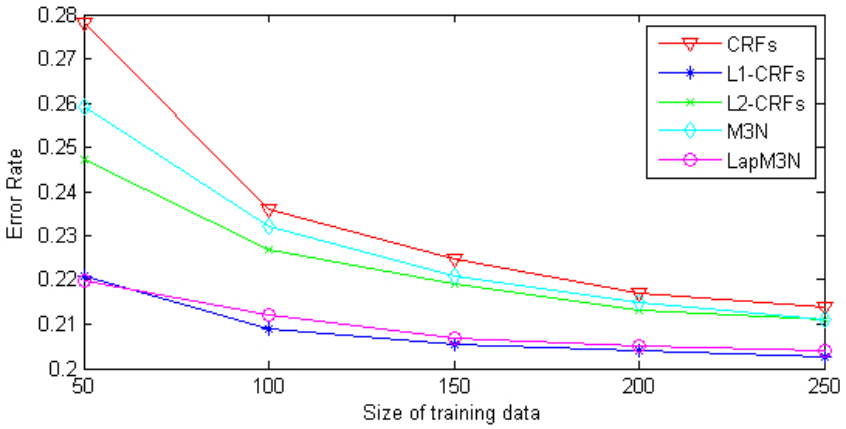

Figure 3: Results on data sets with 30 relevant features.

The only difference is the loss they try to optimize. CRFs optimize the log-loss while M^{3}N optimizes the hinge-loss. As the number of training data increases, all the algorithms consistently get higher performance. The advantage of the LapM^{3}N is more obvious when there are fewer relevant features.

6.1.2 Correlated Features

In reality, most data sets contain redundancy and the features are usually correlated. So, we evaluate our models on synthetic data sets with correlated features. We take the similar procedure as in generating the data sets with i.i.d features to first generate 10 linear-chain CRF model. Then, we use each CRF model to generate one data set of which each sample has 30 relevant features. The 30 relevant features are partitioned into 10 groups. For the features in each group, we first draw a real-value from a standard normal distribution and then 'spoil' the feature with a random Gaussian noise to get 3 correlated features. The noise Gaussian has a zero mean and standard variance 0.05. Here and in all the remaining experiments, we use the sub-gradient method [23] to solve the QP problem in both M^{3}N and LapM^{3}N. We use the learning rate and complexity constant that are suggested by the authors, that is, $\alpha_t = \frac{1}{2\beta\sqrt{t}}$ and $C = 200\beta$, where β is a parameter we introduced to adjust α_t and C. We do K-fold CV on each data set and take the average over the 10 data sets as the final results. Like [25], in each run we choose one part to do training and test on the rest K-1 parts. We vary K from 20, 10, 7, 5, to 4. In other words, we use 50, 100, about 150, 200, and 250 samples during the training. We use the same grid search to choose λ and β from $\{9, 16, 25, 36, 49, 64\}$ and $\{1, 10, 20, 30, 40, 50, 60\}$ respectively. Results are shown in Figure 3. We can get the same conclusions as in the previous results.

Figure 4 shows the average variances of the 100 features, the weights of 200 state feature functions in the model that generates the data, and the average weights of the learned models on the first data set. All the averages are taken over 10 fold cross-validation. We can see that the LapM^{3}N can automatically group the first 30 features into 10 groups and in each group the three features have correlated variances. There is no obvious correlation among

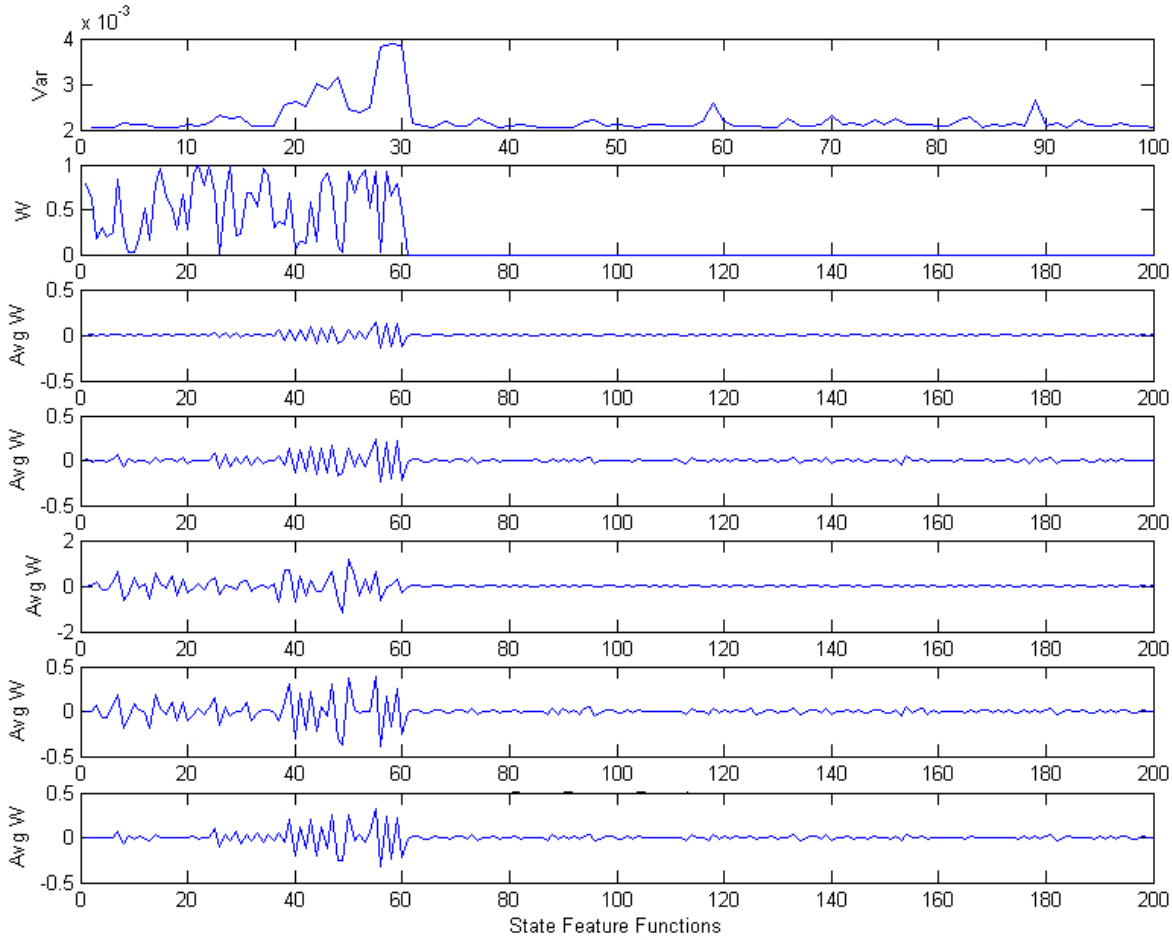

Figure 4: From top to bottom, plot 1 shows the average variances of the features on the first data set in LapM^{3}N; plot 2 shows the weights of the state feature functions in the linear-chain CRF model from which the data are generated; plot 3 to plot 7 show the average weights of the learned LapM^{3}N, M^{3}N, CRFs, L_2-CRFs, and L_1-CRFs over 10 fold CV respectively.

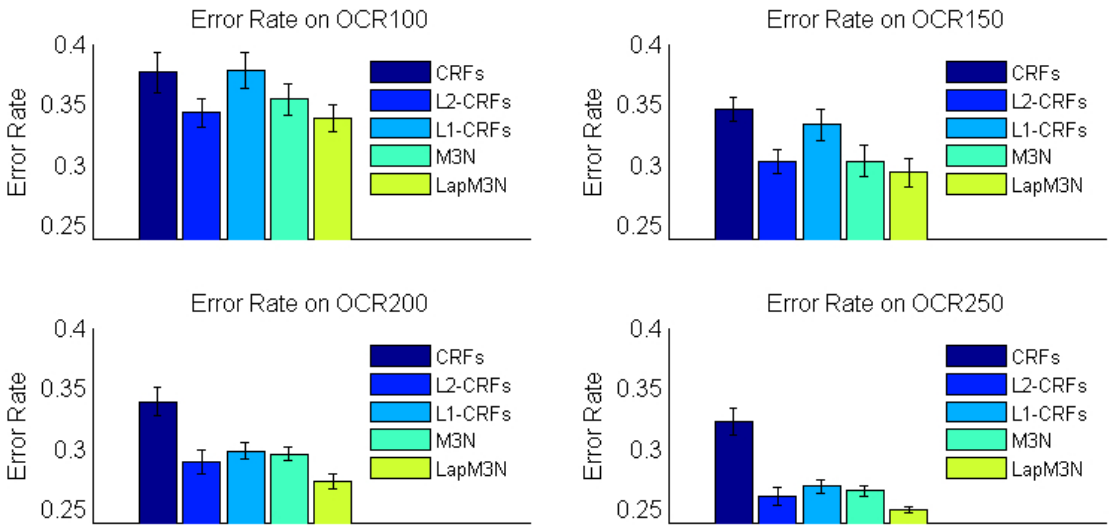

Figure 5: Evaluation results on OCR data set with different numbers of selected data.

other features. For LapM^{3}N and L_1-CRFs, the average weights (posterior means) of the last 140 state feature functions (corresponding to the last 70 irrelevant features) are extremely small. In contrast, CRFs and L_2-CRFs have more larger values of the last 140 state feature functions. For the first 30 relevant features, the LapM^{3}N has a similar plot as the sparse L_1-CRFs with some weights extremely small. Again, CRFs and L_2-CRFs have more feature functions with large average weights. These plots suggest that our LapM^{3}N can recover the sparse data well. Note that all the models have quite different average weights from the model that generates the data. This is because we use a Gibbs sampler to assign labels to the generated data instead of using the labels that are predicted by the model that generate the data. In fact, if we use the model that generates the data to do prediction on its generated data, the error rate is about 0.5. Thus, the learned models, which get lower error rates, are different from the model that generates the data.

6.2 Real-World OCR Data Set

The OCR data set is partitioned into 10 subsets for 10-fold CV as in [25, 23]. We randomly select N samples from each fold and put them together to do 10-fold CV. We vary N from 100, 150, 200, to 250, and denote the selected data sets by OCR100, OCR150, OCR200, and OCR250 respectively. When $\beta = 4$ on OCR100 and OCR150, $\beta = 2$ on OCR200 and OCR250, and $\lambda = 36$, results are shown in Figure 5.

We can see that as the number of training data increases all the algorithms get lower error rates and smaller variances. Generally, the LapM^{3}N consistently outperforms all the other models. M^{3}N outperforms the standard, non-regularized, CRFs and the L_1-CRFs. Again, L_2-CRFs perform comparably with M^{3}N. This is a bit surprising but still reasonable due to the understanding of their only difference on loss functions [11] as we have stated. By examining the prediction accuracy during the learning, we can see an obvious over-fitting in

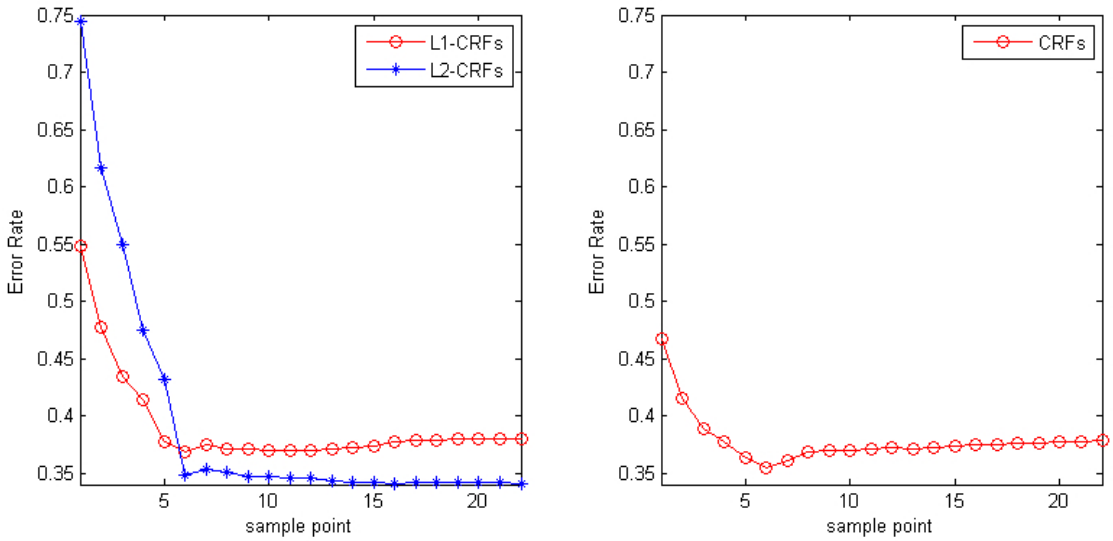

Figure 6: The prediction error rate during the learning at sample points. For the left plot, the sampling points are the relative change ratios of the log-likelihood and from left to right, the change ratios are 1, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.004, 0.003, 0.002, 0.001, 0.0005, 0.0004, 0.0003, 0.0002, 0.0001, and 0.00005; for the right plot, the sampling points are the negative log-likelihoods, and from left to right they are 1000, 800, 700, 600, 500, 300, 100, 50, 30, 10, 5, 3, 1, 0.5, 0.3, 0.1, 0.05, 0.03, 0.01, 0.005, 0.003, and 0.002.

CRFs and L_1-CRFs as shown in Figure 6. In contrast, L_2-CRFs are very robust. This is because unlike the synthetic data sets, features in real-world data are usually not completely irrelevant. In this case, putting small weights to zero as in L_1-CRFs will hurt generalization ability and also lead to instability to regularization constants as shown later. Instead, L_2-CRFs do not put small weights to zero but shrink them towards zero as in the LapM^{3}N. The non-regularized maximum likelihood estimation can easily lead to over-fitting too.

6.3 Sensitivity to Regularization Constants

Figure 7 shows the error rates of different models on the data set OCR100. From the results, we can see that the L_1-CRFs are much sensitive to the regularization constants. However, L_2-CRFs, M^{3}N, and LapM^{3}N are much less sensitive. Among all the models, LapM^{3}N is the most stable one. The stability of LapM^{3}N is due to the posterior weighting instead of hard-thresholding to set small weights to zero as in the L_1-CRFs.

7 Related Work

Our work is directly motivated by the Maximum Entropy Discrimination [12] which is a combination of max-margin methods and Bayesian learning in the single label setting. We

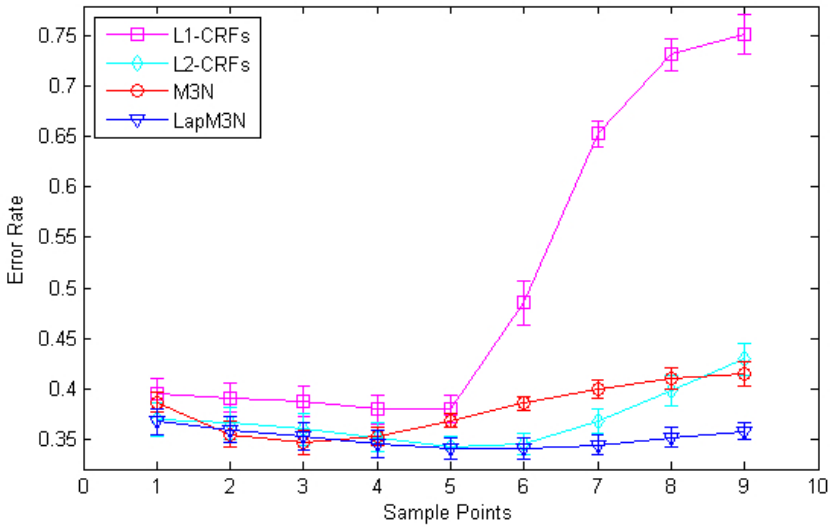

Figure 7: Error rates of different models on OCR100 with different regularization constants. From left to right, the regularization constants are 0.0001, 0.001, 0.01, 0.1, 1, 4, 9, 16, and 25 for L_1-CRFs and L_2-CRFs, and for M^{3}N and LapM^{3}N they are 1, 4, 9, 16, 25, 36, 49, 64, and 81.

present a structured version and under this framework we propose the Bayesian max-margin Markov networks. We show that the standard M^{3}N is a special case of the Bayesian M^{3}N and we also propose the Laplace M^{3}N for sparse learning in high dimensional space.

Sparse Bayesian learning is a framework that has been proposed to find sparse and robust solution to regression and classification. Relevance vector machine (RVM) [28] is studied along that line with kernels. RVM is proposed based on SVM. But unlike SVM which directly optimizes on the margins, RVM defines a likelihood function from the margins with a Gaussian distribution for regression and a logistic sigmoid link function for classification and then does *type-II maximum likelihood* estimation, that is, RVM maximizes the *marginal likelihood*. Although called sparse Bayesian learning [10, 8], as shown in [14] the sparsity is actually due to the MAP estimation. The similar ambiguity of RVM is justified in [31]. We take the full Bayesian approach and optimize a generalized maximum entropy with a set of the *expected* margin constraints. This clarity makes it possible to develop a simple learning algorithm based on existing inference algorithms developed for M^{3}N. Similarly, by defining likelihood functions with margins, Bayesian interpretations of both binary and multi-class SVM are presented in [24, 32].

Based on the hierarchical interpretation of the Laplace prior, a Jeffrey's non-informative second-level hyper-prior is proposed in [10] with an EM algorithm developed to find the MAP estimate. The advantage of the Jeffrey's prior is that it is parameter-free. But as shown in [8, 14], usually no advantage is achieved by using the Jeffrey's hyper-prior compared to the Laplace prior. In [28], a gamma hyper-prior is used in place of the second-level exponential as in the hierarchical interpretation of the Laplace prior.

To encourage sparsity in SVM, two strategies have been used. The first is replacing L_2-

norm by L_1-norm of the weights [4, 33]. The second strategy is to explicitly add a cardinality constraint on the weights. This will lead to a hard non-convex optimization problem. Thus, relaxations [6] must be applied. For Maximum Entropy Discrimination, feature selection is studied in [13] by introducing a set of structural variables. It is straightforward to generalize it to the structured learning case but the resultant learning problem is complex and approximation must be done.

Finally, the distribution $p(\mathbf{w})$ in Theorem 2 has a similar form as that in Bayesian Conditional Random Fields (BCRFs) [20]. The difference is that the normalization factor here is dependent on a set of dual variables, which are estimated in a max-margin manner. In testing, our decision rule can be explicitly expressed as a dot product, and we do not need to take the approximate integration as in BCRFs.

8 Conclusions and Future Work

We proposed a novel framework of *Maximum Entropy Discrimination Markov Networks* (MaxEntNet) for Bayesian max-margin learning in structured prediction. This framework gives rise to a general class of Bayesian M^{3}Ns and subsumes the standard M^{3}N as a spacial case where the predictive model is assumed to be linear and the parameter prior is a standard normal. We show that the adoption of a Laplace prior of the parameter in Bayesian M^{3}N leads to a Laplace M^{3}N that enjoys properties expected from a sparsified Bayesian M^{3}N. Unlike the L_1-regularized maximum likelihood estimation which sets small weights to zeros to achieve sparsity, LapM^{3}N weights the parameters *a posteriori*. Features with smaller weights are shrunk more. This posterior weighting effect makes LapM^{3}N more stable with respect to the magnitudes of the regularization coefficients and more generalizable. We demonstrated that on synthetic data LapM^{3}N can recover the sparse model as well as the sparse L_1-regularized MAP estimation, and on real data sets LapM^{3}N can achieve better performance.

The novel framework of MaxEntNet or Bayesian max-margin Markov networks is extremely general. As for future work, we plan to extend the framework to the learning with missing data, such as semi-supervised learning and learning latent hierarchical models [21, 34]. It is also interesting to study the structure learning under this general framework since it is very natural to encode the field knowledge of model structures as a prior distribution and then apply our framework based on the max-margin principle. We also plan to apply the Laplace M^{3}N to more structured prediction tasks. It would be desirable to apply it to some tasks in which approximate inference must be performed. Since MaxEntNet subsumes M^{3}N as a special case, exploring the kernel methods within the framework of MaxEntNet as in M^{3}N is also an interesting topic.

Acknowledgements

This work was conceived and completed while Jun Zhu is with the SAILING Lab directed by Eric Xing at Carnegie Mellon University, under a visiting scholarship sponsored by the State Scholarship Fund of China. The authors would like to thank Ivo Tsang for inspiring discussions and all members of the SAILING Lab for helpful discussions and technical support. Eric Xing is supported by NSF grants CCF-0523757, DBI-0546594, IIS-0713379, DBI-0640543, and a Sloan Research Fellowship in Computer Science; Jun Zhu and Bo Zhang are supported by the National Natural Science Foundation of China, Grant. No. 60321002; and the National Key Foundation R&D Projects, Grant No. 2004CB318108 and 2007CB311003.

Appendix A. Proof of Theorem 4

We follow the same structure as the proof of PAC-Bayes bound for binary classifier [16] and employ the similar technique to generalize to multi-class problems as in [**?**]. Recall that the output space is $\mathcal{Y}$, and the base discriminant function is $F(\,\cdot\,;\mathbf{w}) \in \mathcal{H} : \mathcal{X} \times \mathcal{Y} \to [-c, c]$, where $c > 0$ is a constant. Our averaging model is specified by $h(\mathbf{x}, \mathbf{y}) = \langle F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \rangle_{p(\mathbf{w})}$. We also define the margin of an example $(\mathbf{x}, \mathbf{y})$ for such a function h as,

$$M(h, \mathbf{x}, \mathbf{y}) = h(\mathbf{x}, \mathbf{y}) - \max_{\mathbf{y}' \neq \mathbf{y}} h(\mathbf{x}, \mathbf{y}'). \tag{11}$$

Thus, the model h makes a wrong prediction on $(\mathbf{x}, \mathbf{y})$ only if $M(h, \mathbf{x}, \mathbf{y}) \leq 0$. Let Q be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\mathcal{D}$ be a sample of N examples independently and randomly drawn from Q. With these definitions, we have the PAC-Bayes Theorem 4. For easy reading, we copy the theorem in the following:

Theorem 4: (PAC-Bayes Bound of MaxEntNet) *Let p_0 be any continuous probability distribution over $\mathcal{H}$ and let $\delta \in (0, 1)$. If $\forall \mathbf{w},\ F(\,\cdot\,;\mathbf{w}) \in \mathcal{H} : \mathcal{X} \times \mathcal{Y} \to [-c, c]$, then with probability at least $1 - \delta$ over random samples $\mathcal{D}$ of Q, for very distribution p over $\mathcal{H}$ and for all margin thresholds $\gamma > 0$:*

$$\Pr_Q(M(h, \mathbf{x}, \mathbf{y}) \leq 0) \leq \Pr_{\mathcal{D}}(M(h, \mathbf{x}, \mathbf{y}) \leq \gamma) + O\Big(\sqrt{\frac{\gamma^{-2} KL(p||p_0) \ln(N|\mathcal{Y}|) + \ln N + \ln \delta^{-1}}{N}}\Big).$$

Here, $\Pr_Q(.)$ stands for $\langle . \rangle_Q$ and $\Pr_{\mathcal{D}}(.)$ stands for the empirical average on $\mathcal{D}$.

Proof: Let m be any natural number. For every distribution p, we independently draw m base models (i.e., discriminant functions) $F_i \sim p$ at random. We also independently draw m variables $\mu_i \sim U([-c, c])$, where U denote the uniform distribution. We define the binary functions $g_i : \mathcal{X} \times \mathcal{Y} \to \{-c, +c\}$ by:

$$g_i(\mathbf{x}, \mathbf{y}; F_i, \mu_i) = 2cI(\mu_i < F_i(\mathbf{x}, \mathbf{y})) - c.$$

With the F_i, μ_i, and g_i, we define $\mathcal{H}_m$ as,

$$\mathcal{H}_m = \big\{ f : (\mathbf{x}, \mathbf{y}) \mapsto \frac{1}{m} \sum_{i=1}^{m} g_i(\mathbf{x}, \mathbf{y}; F_i, \mu_i) | F_i \in \mathcal{H}, \mu_i \in [-c, c] \big\}.$$

We denote the distribution of f over the set $\mathcal{H}_m$ by p^m. For a fixed pair $(\mathbf{x}, \mathbf{y})$, the quantities $g_i(\mathbf{x}, \mathbf{y}; F_i, \mu_i)$ are i.i.d bounded random variables with the mean:

$$\begin{aligned}\langle g_i(\mathbf{x},\mathbf{y};F_i,\mu_i)\rangle_{F_i\sim p,\mu_i\sim U[-c,c]} &= \langle (+c)p[\mu_i \le F_i(\mathbf{x},\mathbf{y})|F_i] + (-c)p[\mu_i > F_i(\mathbf{x},\mathbf{y})|F_i]\rangle_{F_i\sim p} \\ &= \langle \frac{1}{2c}c(c+F_i(\mathbf{x},\mathbf{y})) - \frac{1}{2c}c(c-F_i(\mathbf{x},\mathbf{y}))\rangle_{F_i\sim p} \\ &= h(\mathbf{x},\mathbf{y}).\end{aligned}$$

Therefore, $\langle f(\mathbf{x},\mathbf{y})\rangle_{f\sim p^m} = h(\mathbf{x},\mathbf{y})$. Since $f(\mathbf{x},\mathbf{y})$ is the average over m i.i.d bounded variables, Hoeffding's inequality applies. Thus, for every $(\mathbf{x},\mathbf{y})$,

$$\Pr_{f\sim p^m}[f(\mathbf{x},\mathbf{y}) - h(\mathbf{x},\mathbf{y}) > \xi] \le e^{-\frac{m}{2c^2}\xi^2}.$$

For any two events A and B, we have the inequality,

$$\Pr(A) = \Pr(A,B) + \Pr(A,\bar{B}) \le \Pr(B) + \Pr(\bar{B}|A).$$

Thus, for any $\gamma > 0$ we have

$$\Pr_Q\big[M(h,\mathbf{x},\mathbf{y}) \le 0\big] \le \Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big] + \Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) > \frac{\gamma}{2}|M(h,\mathbf{x},\mathbf{y}) \le 0\big]. \quad (12)$$

Fix $h,\mathbf{x}$, and $\mathbf{y}$, and let $\mathbf{y}'$ achieve the margin in (11). Then, we get

$$M(h,\mathbf{x},\mathbf{y}) = h(\mathbf{x},\mathbf{y}) - h(\mathbf{x},\mathbf{y}'), \text{ and } M(f,\mathbf{x},\mathbf{y}) \le f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}').$$

With these two results, since $\langle f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\rangle_{f\sim p^m} = h(\mathbf{x},\mathbf{y}) - h(\mathbf{x},\mathbf{y}')$, we can get

$$\begin{aligned}\Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) > \frac{\gamma}{2}|M(h,\mathbf{x},\mathbf{y}) \le 0\big] &\le \Pr_Q\big[f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}') > \frac{\gamma}{2}|M(h,\mathbf{x},\mathbf{y}) \le 0\big] \\ &\le \Pr_Q\big[f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}') - M(h,\mathbf{x},\mathbf{y}) > \frac{\gamma}{2}\big] \\ &\le e^{-\frac{m\gamma^2}{32c^2}}, \end{aligned} \quad (13)$$

where the first two inequalities are due to the fact that if two events $A \in B$, then $p(A) \le p(B)$, and the last inequality is due to the Hoeffding's inequality.

Substitute (13) into (12), and we get,

$$\Pr_Q\big[M(h,\mathbf{x},\mathbf{y}) \le 0\big] \le \Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big] + e^{-\frac{m\gamma^2}{32c^2}}.$$

Since the left hand side does not depend on f, we take the expectation over $f \sim p^m$ on both sides and get,

$$\Pr_Q\big[M(h,\mathbf{x},\mathbf{y}) \le 0\big] \le \langle \Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big]\rangle_{f\sim p^m} + e^{-\frac{m\gamma^2}{32c^2}}. \quad (14)$$

Let p_0^m be a prior distribution on $\mathcal{H}_m$. p_0^m is constructed from p_0 over $\mathcal{H}$ exactly as p^m is constructed from p. Then, $KL(p^m||p_0^m) = mKL(p||p_0)$. By PAC-Bayes theorem [19], with probability at least $1-\delta$ over sample $\mathcal{D}$, the following bound holds for any distribution p,

$$\begin{aligned}\langle \Pr_Q\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big]\rangle_{f\sim p^m} &\le \langle \Pr_{\mathcal{D}}\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big]\rangle_{f\sim p^m} \\ &\quad + \sqrt{\frac{mKL(p||p_0) + \ln N + \ln \delta^{-1} + 2}{2N-1}}. \end{aligned} \quad (15)$$

By the similar statement as in (12), for every $f \in \mathcal{H}_m$ we have,

$$\Pr_{\mathcal{D}}\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2}\big] \le \Pr_{\mathcal{D}}\big[M(h,\mathbf{x},\mathbf{y}) \le \gamma\big] + \Pr_{\mathcal{D}}\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2} | M(h,\mathbf{x},\mathbf{y}) > \gamma\big]. \quad (16)$$

By rewriting the second term on the right-hand side of (16), we get

$$\begin{aligned}
\Pr_{\mathcal{D}}\big[M(f,\mathbf{x},\mathbf{y}) \le \frac{\gamma}{2} | M(h,\mathbf{x},\mathbf{y}) > \gamma\big] &= \Pr_{\mathcal{D}}\big[\exists \mathbf{y}' \ne \mathbf{y} : \Delta f(\mathbf{x},\mathbf{y}') \le \frac{\gamma}{2} | \forall \mathbf{y}' \ne \mathrm{y} : \Delta h(\mathbf{x},\mathbf{y}') > \gamma\big] \\
&\le \Pr_{\mathcal{D}}\big[\exists \mathbf{y}' \ne \mathbf{y} : \Delta f(\mathbf{x},\mathbf{y}') \le \frac{\gamma}{2} | \Delta h(\mathbf{x},\mathbf{y}') > \gamma\big] \\
&\le \sum_{\mathbf{y}' \ne \mathbf{y}} \Pr_{\mathcal{D}}\big[\Delta f(\mathbf{x},\mathbf{y}') \le \frac{\gamma}{2} | \Delta h(\mathbf{x},\mathbf{y}') > \gamma\big] \\
&\le (|\mathcal{Y}| - 1) e^{-\frac{m\gamma^2}{32c^2}}, \quad (17)
\end{aligned}$$

where we have use $\Delta f(\mathbf{x},\mathbf{y}')$ to denote $f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')$, and use $\Delta h(\mathbf{x},\mathbf{y}')$ to denote $h(\mathbf{x},\mathbf{y}) - h(\mathbf{x},\mathbf{y}')$.

Put (14), (15), (16), and (17) together, then we get following bound holding for any fixed m and $\gamma > 0$,

$$\Pr_{Q}\big[M(h,\mathbf{x},\mathbf{y}) \le 0\big] \le \Pr_{\mathcal{D}}\big[M(h,\mathbf{x},\mathbf{y}) \le \gamma\big] + |\mathcal{Y}| e^{-\frac{m\gamma^2}{32c^2}} + \sqrt{\frac{mKL(p||p_0) + \ln N + \ln \delta^{-1} + 2}{2N-1}}.$$

To finish the proof, we need to remove the dependence on m and γ. This can be done by applying the union bound. By the definition of f, it is obvious that if $f \in \mathcal{H}_m$ then $f(\mathbf{x},\mathbf{y}) \in \{(2k-m)c/m : k = 0, 1, \ldots, m\}$. Thus, even though γ can be any positive value, there are no more than $m+1$ events of the form $\{M(f,\mathbf{x},\mathbf{y}) \le \gamma/2\}$. Since only the application of PAC-Bayes theorem in (15) depends on (m,γ) and all the other steps are true with probability one, we just need to consider the union of countably many events. Let $\delta_{m,k} = \delta/(m(m+1)^2)$, then the union of all the possible events has a probability at most $\sum_{m,k} \delta_{m,k} = \sum_m (m+1)\delta/(m(m+1)^2) = \delta$. Therefore, with probability at least $1-\delta$ over random samples of $\mathcal{D}$, the following bound holds for all m and all $\gamma > 0$,

$$\begin{aligned}
\Pr_{Q}\big[M(h,\mathbf{x},\mathbf{y}) \le 0\big] - \Pr_{\mathcal{D}}\big[M(h,\mathbf{x},\mathbf{y}) \le \gamma\big] &\le |\mathcal{Y}| e^{-\frac{m\gamma^2}{32c^2}} + \sqrt{\frac{mKL(p||p_0) + \ln N + \ln \delta_{m,k}^{-1} + 2}{2N-1}} \\
&\le |\mathcal{Y}| e^{-\frac{m\gamma^2}{32c^2}} + \sqrt{\frac{mKL(p||p_0) + \ln N + 3\ln \frac{m+1}{\delta} + 2}{2N-1}}
\end{aligned}$$

Setting $m = \lceil 16c^2\gamma^{-2} \ln \frac{N|\mathcal{Y}|^2}{KL(p||p_0)+1} \rceil$ gives the results in the theorem. □

Appendix B. Duality of L_1-M^{3}N

Based on L_1-SVM [4], a straightforward formulation of L_1-M^{3}N is as follows,

$$\min_{\mathbf{w},\xi} \ \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{N}\xi_i$$
$$\text{s.t.} \ \ \mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i, \forall i, \ \forall \mathbf{y} \neq \mathbf{y}^i$$

where $\|.\|$ is the L_1-norm, and $\Delta\mathbf{f}_i(\mathbf{y}) = \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \mathbf{f}(\mathbf{x}^i, \mathbf{y})$. $\Delta\ell_i(\mathbf{y})$ is a loss function.

To derive the convex dual problem, we introduce a dual variable $\alpha_i(\mathbf{y})$ for each constraint and form the Lagrangian as follows,

$$L(\alpha, \mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{N}\xi_i - \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\big(\mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y}) - \Delta\ell_i(\mathbf{y}) + \xi_i\big).$$

By definition, the Lagrangian dual is,

$$\begin{aligned}
L^\star(\alpha) &= \inf_{\mathbf{w},\xi} L(\alpha, \mathbf{w}, \xi) \\
&= \inf_{\mathbf{w}}\Big[\frac{1}{2}\|\mathbf{w}\| - \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y})\Big] + \inf_{\xi}\Big[C\sum_{i=1}^{N}\xi_i - \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\xi_i\Big] + \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\ell_i(\mathbf{y}) \\
&= -\sup_{\mathbf{w}}\Big[\mathbf{w}^\top\big(\sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\mathbf{f}_i(\mathbf{y})\big) - \frac{1}{2}\|\mathbf{w}\|\Big] - \sup_{\xi}\Big[\sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\xi_i - C\sum_{i=1}^{N}\xi_i\Big] + \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\ell_i(\mathbf{y}).
\end{aligned}$$

Again, by definition, the first term in the right-hand side is the convex conjugate of $\phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|$ and the second term is the conjugate of $U(\xi) = \sum_{i=1}^{N}\xi_i$. It is easy to show that,

$$\phi^\star(\alpha) = \mathbb{I}_\infty(|\sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\mathbf{f}_i^k(\mathbf{y})| \leq \frac{1}{2}, \ \forall 1 \leq k \leq K),$$

and

$$U^\star(\alpha) = \mathbb{I}_\infty(\sum_{\mathbf{y}}\alpha_i(\mathbf{y}) \leq C, \ \forall i),$$

where as defined before $\mathbb{I}_\infty(\cdot)$ is an indicator function that equals zero when its argument is true and infinity otherwise. $\Delta\mathbf{f}_i^k(\mathbf{y}) = f_k(\mathbf{x}^i, \mathbf{y}^i) - f_k(\mathbf{x}, \mathbf{y})$.

Therefore, we get the dual problem as follows,

$$\begin{aligned}
\max_{\alpha} \quad & \sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\ell_i(\mathbf{y}) \\
\text{s.t.} \quad & |\sum_{i,\mathbf{y}}\alpha_i(\mathbf{y})\Delta\mathbf{f}_i^k(\mathbf{y})| \leq \frac{1}{2}, \ \forall k \\
& \sum_{\mathbf{y}}\alpha_i(\mathbf{y}) \leq C, \ \forall i.
\end{aligned}$$

References

[1] Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. Hidden markov support vector machines. In *International Conference on Machine Learning*, 2003.

[2] Galen Andrew and Jianfeng Gao. Scalable training of l_1-regularized log-linear models. In *International Conference on Machine Learning*, 2007.

[3] P. Bartlett, M. Collins, B. Taskar, and D. McAllester. Exponentiated gradient algorithms for larg-margin structured classification. In *Advances in Neural Information Processing Systems*, 2004.

[4] Kristin P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optim. Methods Softw*, (1):23–34, 1992.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] Antoni B. Chan, Nuno Vasconcelos, and Gert R. G. Lanckriet. Direct convex relaxations of sparse svm. In *International Conference on Machine Learning*, 2007.

[7] Mirosla Dudík, Steven J. Phillips, and Robert E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, (8):1217–1260, 2007.

[8] Susana Eyheramendy, Alexander Genkin, Wen-Hua Ju, David D. Lewis, and David Madiagan. Sparse bayesian classifiers for text categorization. Technical report, Rutgers University, 2003.

[9] Mario Figueiredo. Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1150–1159, 2003.

[10] Mario A. T. Figueiredo. Adaptive sparseness using jeffreys prior. In *Advances in Neural Information Processing Systems*, 2001.

[11] Amir Globerson, Terry Y. Koo, Xavier Carreras, and Michael Collins. Exponentiated gradient algorithms for log-linear structured prediction. In *International Conference on Machine Learning*, 2007.

[12] Tommi Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems*, 1999.

[13] Tony Jebara and Tommi Jaakkola. Feature selection and dualities in maximum entropy discrimination. In *Uncertainty in Artificial Intelligence*, 2000.

[14] Ata Kaban. On bayesian classification with laplace priors. *Pattern Recognition Letters*, 28(10):1271–1282, 2007.

[15] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[16] John Langford, Matthias Seeger, and Nimrod Megiddo. An improved predictive accuracy bound for averaging classifiers. In *International Conference on Machine Learning*, 2001.

[17] Guy Lebanon and John Lafferty. Boosting and maximum likelihood for exponential models. In *Advances in Neural Information Processing Systems*, 2001.

[18] Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l_1-regularization. In *Advances in Neural Information Processing Systems*, 2006.

[19] David McAllester. Pac-bayesian model averaging. In *the Twelfth Annual Conference on Computational Learning Theory*, 1999.

[20] Yuan (Alan) Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. In *International Conference on Artificial Intelligence and Statistics*, 2005.

[21] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*, 2004.

[22] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, (77(2)):257–286, 1989.

[23] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. (online) subgradient methods for structured prediction. In *International Conference on Artificial Intelligence and Statistics*, 2007.

[24] Peter Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Journal of Machine Learning Research*, (46):21–52, 2002.

[25] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems*, 2003.

[26] Ben Taskar, Simon Lacoste-Julien, and Michael I. Jordan. Structured prediction via the extragradient method. In *Advances in Neural Information Processing Systems*, 2006.

[27] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc.*, B(58):267–288, 1996.

[28] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, (1):211–244, 2001.

[29] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, 2004.

[30] Martin J. Wainwright, Pradeep Ravikumar, and John Lafferty. High-dimensional graphical model selection using l1-regularized logistic regression. In *Advances in Neural Information Processing Systems*, 2006.

[31] David Wipf, Jason Palmer, and Bhaskar Rao. Perspectives on sparse bayesian learning. In *Advances in Neural Information Processing Systems*, 2003.

[32] Zhihua Zhang and Michael I. Jordan. Bayesian multicategory support vector machines. In *Uncertainty in Artificial Intelligence*, 2006.

[33] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems*, 2004.

[34] Jun Zhu, Zaiqing Nie, Bo Zhang, and Ji-Rong Wen. Dynamic hierarchical markov random fields and their application to web data extraction. In *International Conference on Machine Learning*, 2007.